第2章 マスタ・シリアル・モードの詳細とトラブル対策 FPGAのコンフィグレーション基礎知識《Xilinx編》

安井 健

ここでは，米国Xilinx社のFPGAのうち，Virtex-4/5とSpartan-3/E/A/AN/A DSPにおけるコンフィグレーションについて解説する．特に，専用コンフィグレーションROMを使う際に用いるマスタ・シリアル・モードについて詳しく扱う．また，コンフィグレーションが正常に動作しないといったトラブル発生時の解決法を，症状別に説明する．　（編集部）

SRAMベースのFPGA（Field Programmable Gate Array）では，電源投入後にユーザ回路をFPGAへ書き込む，コンフィグレーションと呼ばれる作業が必要です．SRAMベースのFPGAを使用するすべてのユーザがコンフィグレーションを行う必要がありますが，基板上の構成や実行されているプロセスについてはあまり気にせず使っている方が多いのではないでしょうか．どんなに素晴らしい回路設計を行ってもコンフィグレーションがうまくいかないとFPGAは動作しません．FPGAを使うに当たって，ユーザにとっては重要な項目であると思います．

本稿では，米国Xilinx社のFPGAを対象としてコンフィグレーションに関する動作説明とトラブル発生時の対策について説明します．

1. さまざまなコンフィグレーション・モード

Xilinx社のFPGAであるVirtex-4/5とSpartan-3 Generation（Spartan-3/3E/3A/3AN/3A DSPの総称）ファミリでサポートしているコンフィグレーション・モードを**表1**に示します．

● 3本のピンでモードを指定する

FPGAに電源が投入されると，FPGAが持つ3本のモード・ピン（M2，M1，M0）に入力されている論理値によってコンフィグレーションを行う方法（コンフィグレーション・モード）が決まります．

マスタ・シリアルとスレーブ・シリアル，JTAGの三つのモードの設定値は，すべてのファミリで共通です．

コンフィグレーション・モードは，以下のように分類できます．

- マスタとスレーブのモード
- シリアルとパラレルのモード
- JTAGモード
- 汎用フラッシュ・メモリ

● マスタとスレーブ

マスタ・モードは，コンフィグレーション・データ（ビットストリームと呼ぶ）を受け取るタイミングを自ら生成します．コンフィグレーションに必要なクロックをFPGAが出力します．

スレーブ・モードは，クロックとコンフィグレーション・データの両方を外部からもらいます．

● シリアルとパラレル

シリアル・モードは，コンフィグレーション・データを1ビットずつ受信します．

KeyWord
FPGA，Virtex 4/5，Spartan-3 Generation，コンフィグレーション，JTAG，SPI，BPI，マスタ・シリアル・モード

表1　コンフィグレーション・モード

モード・ピン (M2,M1,M0)	Virtex-5	Virtex-4	Spartan-3A /3AN/3A DSP	Spartan-3E	Spartan-3
000	マスタ・シリアル				
001	マスタSPI	スレーブ・セレクト・マップ（32ビット）	マスタSPI		—
010	マスタBPI（Up）（8ビットまたは16ビット）	—	マスターBPI（Up）（8ビット）		—
011	マスタBPI（Down）（8ビットまたは16ビット）	マスタ・セレクト・マップ	内部SPI[1]（Spartan-3ANのみ）	マスタBPI（Down）（8ビット）	マスタ・セレクト・マップ
100	マスタ・セレクト・マップ（8ビット，16ビット自動認識）	—	—	—	—
101	JTAG				
110	スレーブ・セレクト・マップ（8ビット，16ビット，32ビット自動認識）	スレーブ・セレクト・マップ（8ビット）	スレーブ・セレクト・マップ		
111	スレーブ・シリアル				

注：BPIモードには，メモリのアドレスをインクリメントしながらデータを読み出すUpモードと，アドレスをデクリメントしながらデータを読み出すDownモードがある．

パラレル・モードは，複数のビットを一つの単位として受信します．パラレル・モードをセレクト・マップ・モードと呼ぶ場合もあります．

セレクト・マップでは，基本的には8ビット単位でコンフィグレーションを行います．Virtex-4とVirtex-5では，さらに16ビット幅と32ビット幅のモードも用意されています．

セレクト・マップでは，コンフィグレーションのために多くのピンを使用します．FPGAにとって必要なデータ量は，コンフィグレーションのモードで変わりません．つまり，データをパラレルで受信すれば，シリアルで受信するよりもコンフィグレーション時間が短くて済みます．

Virtex-4のセレクト・マップでは，ビット幅ごとにモード・ピンの設定値が規定されています．Virtex-5では，ビットストリームの先頭にバス幅検出用のデータが含まれており，データ・バスのどのバイト・レーンにバス幅検出用データが出現するかを自動的に判別することでコンフィグレーション時のビット幅が決まります．

● JTAGモード

JTAGモードは，パソコンからダウンロード・ケーブルを使用してFPGAを直接コンフィグレーションする場合に使うことが多い方式です．Xilinx社では，System ACEというFPGAのJTAGを使用しています．System ACEは，CompactFlashからコンフィグレーションするシステムです．また，ChipScopeのようなオンチップ・デバッグ・ツールでは，JTAG経由でFPGA内部へアクセスします．

デバッグの用途などを考えると，FPGAのJTAGピンへアクセスできるように基板を設計しておくことをお勧めします．

● 汎用フラッシュ・メモリからのコンフィグレーション

Spartan-3E以降のファミリ（Spartan-3E/3A/3AN/3A DSP，Virtex-5）では，汎用フラッシュ・メモリから直接コンフィグレーションできるようになりました．SPI（Serial Peripheral Interface）のシリアル・タイプとBPI（Byte Peripheral Interface）のパラレル・フラッシュの両方に対応しています．

Spartan-3までは，汎用フラッシュ・メモリからコンフィグレーションする場合，フラッシュ・メモリとFPGAの間に制御用の回路（CPLDを使うことが多い）を配置する必要がありました．これに対し，Spartan-3E以降では，データシートなどに記載されているフラッシュ・メモリであればFPGAと直接接続できます．

Spartan-3A系の三つのファミリ（Spartan-3A/3AN/3A DSP）では，基本的に同じコンフィグレーション・モードが使用可能です．Spartan-3ANにのみ，内部SPIモードがあります．

Spartan-3ANは，SRAMベースのFPGAとコンフィグレーション用のフラッシュ・メモリを1チップにしたマルチチップ・モジュールです．内蔵のコンフィグレーション用フラッシュ・メモリを使うときには，内部SPIモードを

用います．この際，V_{CCAUX}電源に3.3Vが必要である点に注意が必要です．

● 汎用フラッシュ・メモリを使用したマルチブート

汎用フラッシュ・メモリをコンフィグレーション・データの保存場所として使用するメリットの一つとして，保存可能なデータ容量が大きい点が挙げられます．あらかじめ複数の回路（コンフィグレーション・データ）を用意しておき，動作状態に合わせて切り替えながら使うようなアプリケーションや，フィールドでアップデートを行いたい場合などに活用できます．

汎用フラッシュ・メモリからのコンフィグレーションをサポートしているSpartan-3E以降のファミリでは，あらかじめ複数のコンフィグレーション・データを使用することを考慮した，マルチブート構成をとることができます．

マルチブートに関する特徴を**表2**に，使用するマクロを**図1**に示します．これらのマクロをユーザ回路の一部として組み込んでおき，制御することで，マルチブートが可能になります．

Spartan-3Eでは，一つのフラッシュ・メモリにコンフィグレーション・データを2個までしか搭載できません．しかしコンフィグレーション・データを切り替えるトリガのかけ方は簡単です．ユーザ回路の一部にSTARTUP_SPARTAN3Eセルを組み込んでおき，そのMBT（Multi Boot Triger）入力ピンに300ns以上の“L”パルスを入れることで再コンフィグレーションがかかります．

Spartan-3AとVirtex-5では，ユーザ回路の一部にICAP（Internal Configuration Access Port）セルを組み込み，入力データ用バスから一連のコンフィグレーション・コマンドを入力する必要があります．シーケンスを与える必要がある点で，若干手順が複雑ですが，リブート先の読み出しアドレスを任意に選択できるなど自由度があります．コンフィグレーション専用のウォッチドッグ・タイマが搭載されているSpartan-3Aでは，フラッシュ・メモリへの書き込みが不完全であった場合などを想定して，マルチブート・トリガ後の規定時間内に期待するビットストリームが受信できない場合に，再度デフォルトのアドレスのデータでコンフィグレーションし直すことが自動でできます．

● コンフィグレーション・データのファイル形式

コンフィグレーションで使用する代表的なファイル形式を**表3**に示します．

専用コンフィグレーションROMに書き込むためのmcsファイルと，JTAGでFPGAへ直接ダウンロードする際に使用するbitファイルが最もよく使われています．

最近，組み込みシステムで，外部からコンフィグレー

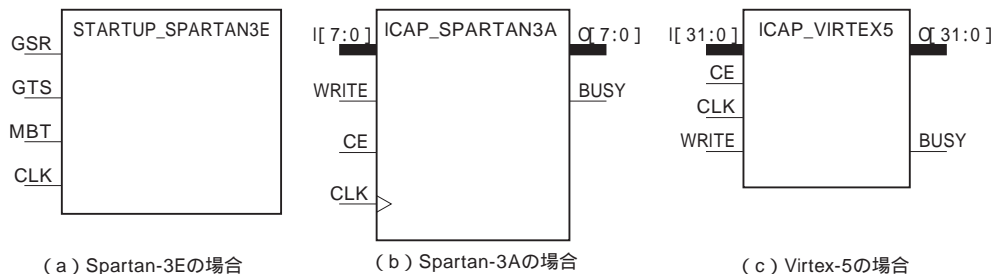
表2 マルチブートの対応

	Spartan-3E	Spartan-3A	Virtex-5
マルチブート手順の複雑さ	簡単	若干複雑	若干複雑
BPIによるマルチブート	可[注1]	可[注2]	可[注2]
SPIによるマルチブート	不可	可	可
マルチブートのトリガ方法	STARTUPプリミティブのMBTピンに‘L’パルス入力	ICAPプリミティブにリブート・シーケンスを入力	ICAPプリミティブにリブート・シーケンスを入力
使用可能なビット・データ数	2	フラッシュのサイズによる	フラッシュのサイズによる
ウォッチドッグ・タイマ	なし	あり	あり

注1　リブート時はモード・ピンの設定と反対方向から再コンフィグレーションされる．例えばBPI-UPモードに基板上設定されている場合，再コンフィグはBPI-DOWNで行われる．

注2　マルチブート時の開始アドレスはリブート前のユーザ回路が指定する．

（a）Spartan-3Eの場合　（b）Spartan-3Aの場合　（c）Virtex-5の場合

図1 マルチブートに使用するマクロ

ションROMのデータを書き換え，リモート・アップデートを行いたいといった要望を聞くことがあります．そのような場合にはXSVFやACEファイルが活用できます．

2．マスタ・シリアル・モードを使いこなす

FPGAは，さまざまなコンフィグレーション・モードを持ちますが，最も基本となるのはマスタ・シリアル・モードといえます．ここでは，マスタ・シリアル・モードにおけるコンフィグレーション回路/基板設計上の注意点について説明します．

Spartan-3Aをマスタ・シリアル・モードで専用コンフィグレーションROM（Platform Flash PROM）からコンフィグレーションする際のコンフィグレーション回路を**図2**に示します．

Spartan-3Aの電源系はV_{CCINT}，V_{CCAUX}，V_{CCO}（4バンク分）の3系統ありますが，V_{CCAUX}は2.5Vまたは3.3Vのどちらかから選択できるようになりました．**図2**はV_{CCAUX}を2.5Vとした場合の接続で，この場合Spartan-3Eでも同様の接続が可能です．

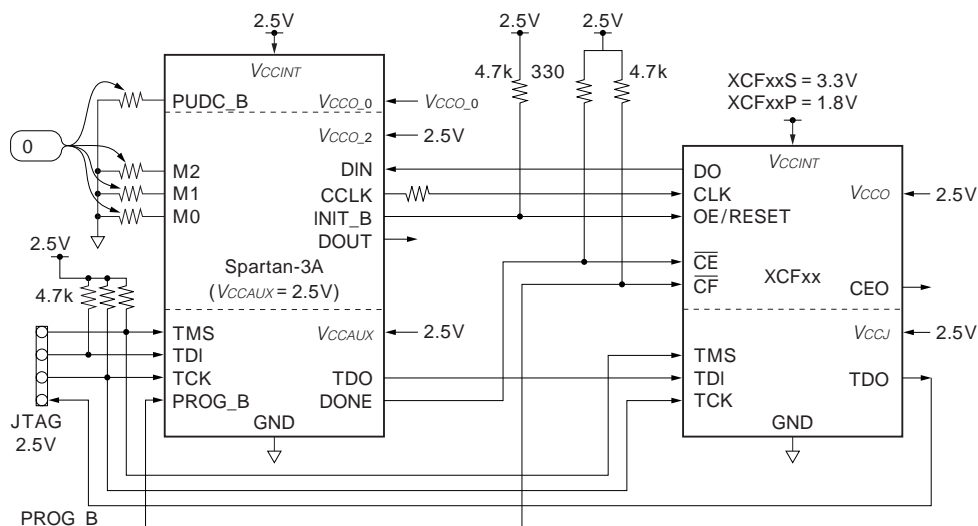
● 電源

プログラム・ピン，JTAGピン，DONEピンはV_{CCAUX}電源系に含まれています．このため，これらのインターフェース用のI/O電圧はV_{CCAUX}の電圧に合わせておく必要

表3 コンフィグレーションで使用する代表的なファイル

拡張子	説　明
.BIT	iMPACTでFPGAへJTAG経由でダウンロードする際に使用する．バイナリ・ファイルでコンフィグレーション・データのほかにヘッダ情報を含む．
.BIN	.BITファイルからコンフィグレーション・データ部分のみを取り出したバイナリ・ファイル．マイクロプロセッサなどからユーザ独自のコンフィグレーション方法を行う際に利用可能．Bitgenのオプション設定により生成する．
.MCS	コンフィグレーションROM書き込み用に使用するテキスト形式ファイル．コンフィグレーション・データのほかに，アドレス，チェック・サム情報を含む．iMPACTで専用コンフィグレーションROM（Platform Flash PROM）をターゲットに生成した場合，自動的にビット・スワップが行われている．SPIフラッシュ・メモリをターゲットにした場合はビット・スワップは行われない．
.HEX	コンフィグレーション・データのみを含むテキスト形式ファイル．ユーザ独自のコンフィグレーション方法を行う際に利用する．iMPACTで生成する際にビット・スワップの有無はユーザが選択可能．
.XSVF	Xilinx Serial Vector Formatの略．JTAGの操作手順記述には，一般的には，SVFファイルが使用されるが，テキスト形式の.SVFのファイル・サイズを縮小する目的でXilinx社が独自に定義したバイナリ・ファイルのフォーマット．アプリケーション・ノートXapp058，Xapp503にフォーマットの記載があり，特に組み込みシステムでコンフィグレーションROMの書き換えを行うような場合に使用可能．
.ACE	主にSystem ACEコントローラを使用したCompactFlashからJTAGでコンフィグレーションするためのデータ・フォーマット．Xapp424にACEファイルを仕様したJTAGコントローラの記載がある．

図2 マスタ・シリアル・モードの回路例

Spartan-3AにおいてV_{CCAUX}＝2.5Vの場合を示す．図には記載していないが，パワー・マネージメント機能のサスペンド・モードを使用しない場合は，SUSPENDピンをGNDに接続する．

があります．また，JTAG用ピン・ヘッダ部にある電圧参照用のV_{REF}電圧と，DONEピンのプルアップ電圧もV_{CCAUX}に合わせてください．

専用コンフィグレーションROMとのインターフェースには，I/Oバンク2を使用します．バンク2のユーザI/Oで使用するI/O電圧V_{CCO_2}を決めたら，専用コンフィグレーションROMのV_{CCO}電圧をV_{CCO_2}に合わせておきます．また，PUDC_Bピンはバンク0に含まれます．PUDC_Bに“H”レベルを入力する場合は，V_{CCO_0}の電圧レベルを入れることになります．

Spartan-3Aのパワー ONリセットは，V_{CCINT}，V_{CCAUX}，V_{CCO_2}の電圧が規定値まで上昇すると解除されます．パワーONリセットに使用されるI/O電源のバンクは，デバイス・ファミリにより異なります．どのFPGAでもパワーONリセットに使用される電源は規定の時間内に単調増加で上昇させる必要があります．電源投入直後はFPGAや周辺チップ用のバイパス・コンデンサが充電されたり，FPGA自体パワーONに一定量の電流が必要となるため，完全な単調増加が難しい場合もありますが，**図3**のようなイメージで電源投入のOK/NGを判断できます．**図3(a)**は理想的な波形で，規定時間内に推奨電圧まで滑らかに電源が上昇しています． **図3(b)**は途中で上昇が緩やかになっていますがドロップするまでは至っていないためOKと判断します．**図3(c)**はドロップが発生しているため修正が必要です．

● プログラム・ピン(PROG_B)

プログラム・ピンに“L”レベルを入力することでコンフィグレーション・メモリをクリアします．

電源投入時の電圧が上昇している最中は，プログラム・ピンを“L”固定にしてコンフィグレーションが始まらないようにし，電源が安定した後“H”へ遷移させることで安定したコンフィグレーションを行えます．このため，プログラム・ピンの制御には電源監視ICやリセットICを使用することを推奨しています．

なお，プログラム・ピンにはV_{CCAUX}電源への内部プルアップがあります．ピン自体が未接続や基板上で単純なプルアップになっていても特に問題ありませんが，できるだけプログラム・ピンを制御可能なように構成します．

● INIT_Bピン

INIT_Bピンはオープン・ドレインです．コンフィグレーション・メモリのクリア中はFPGAが“L”へドライブし，完了するとハイ・インピーダンス(Hi-Z)になります．基板上には4.7kΩのプルアップを付けておくことで，メモリ・クリア完了後外部からば“H”出力が見えるようになります．

“H”が確認されると次のプロセスへ移行しますが，この段階でコンフィグレーションの開始を遅らせたい場合は，外部から“L”にドライブできます．プルアップ抵抗は1kΩ程度の小さな値でも問題ありませんが，4.7kΩを一般的には勧めています．

INIT_Bピンを専用コンフィグレーションROMのアクティブ“L”のリセット・ピンに接続することで，FPGAがメモリ・クリア中は専用コンフィグレーションROMもリセット状態になります．FPGAのメモリ・クリアが終了すると専用コンフィグレーションROMもリセット解除される動作となります．

(a) 理想的な電源立ち上がり

(b) 許容範囲内の電源立ち上がり

(c) 修正が必要な電源立ち上がり

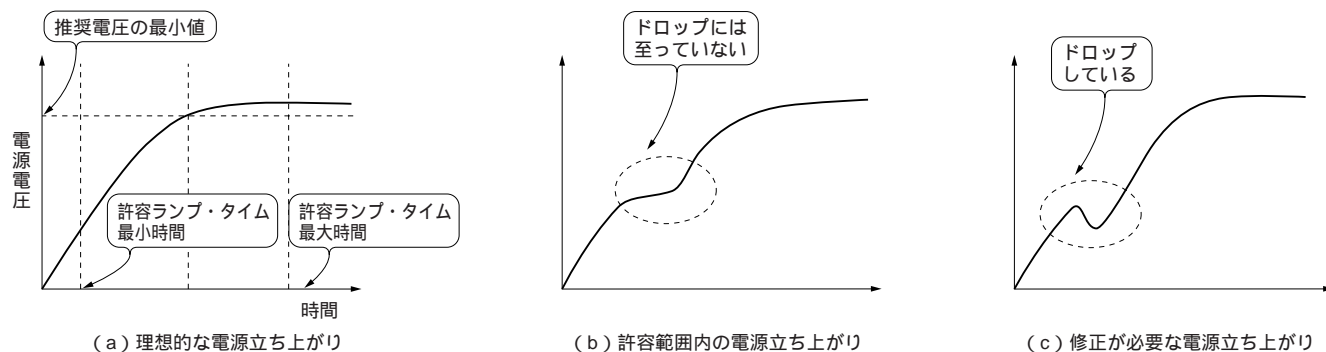



図3 電源の立ち上がり
ドロップが発生している場合は修正が必要．

● モード・ピン(M[2：0])

モード・ピンはすべてのデバイスで3ピンあり，コンフィグレーション・モードを決めるピンです．

INIT_Bの立ち上がりエッジで値が読み取られ，コンフィグレーション・モードが決定します．INIT_Bの立ち上がり時にはモード・ピンのレベルを完全に確定しておきたいため，GNDまたはV_{CCO_2}の電圧レベルに直接，接続します．

デバッグ時などは，例えばJTAGモードに変更したい場合などを想定して，0Ω抵抗を介して接続することをお勧めします．ただし，Spartan-3E/3Aにおいて，コンフィグレーション後にモード・ピンをユーザI/Oとして使用する場合は，0Ωで接続することができないので注意が必要です．

● PUDC_Bピン

PUDC_BはSpartan-3AではPullup During Configurationという意味です．ほかのファミリではHSWAP_ENやHSWAPという名称が使われています．呼び方が違っていても機能は変わりません．

このピンはコンフィグレーション中にユーザI/Oに内部プルアップを適用するか，ハイ・インピーダンスのままにするかを選択するピンです．コンフィグレーション中は値を固定しておく必要があります．

モード・ピンと同様に，ユーザI/Oとして使用しない場合は0Ω抵抗を介してGNDや電源に接続しておくことをお勧めします．

図2の例では，GNDへ接続しています．こうすることで，コンフィグレーション中にV_{CCO}への内部プルアップが働いて，ユーザI/Oは外部からは“H”レベルに見えます．

PUDC_Bで設定できるコンフィグレーション中のI/Oピンの状態は，ハイ・インピーダンスか内部プルアップによる“H”レベルのどちらかです．しかし，どうしてもコンフィグレーション中に“L”レベルをユーザI/Oに出したい場合もあるでしょう．このような場合は，PUDC_Bを“L”にしてI/Oを“H”レベルにしてインバータを外付けするか，基板上にプルダウン抵抗を配置することになります．PUDC_Bが“H”でユーザI/Oがハイ・インピーダンスの場合はプルダウン抵抗値の選択にあまり気を使う必要はありません．PUDC_Bが“L”で内部プルアップが有効になっている場合は，内部抵抗に打ち勝つようにかなり小さめの値の外部プルダウン抵抗を選択する必要があります．

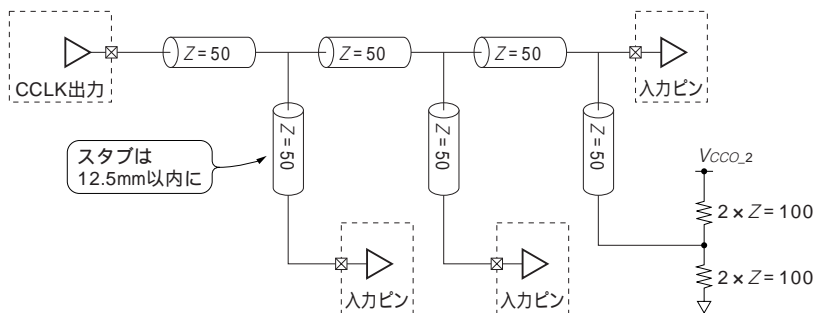
● CCLKピン

CCLKは，コンフィグレーション用のクロック・ピンです．INIT_Bの立ち上がりエッジでモード・ピンがサンプルされ，マスタ・モードであることが認識されるとFPGAはCCLKのトグルを始めます．

Spartan-3A以外のFPGAファミリでは，マスタ・モードFPGAのCCLKは双方向ピンとして構成されます．このため，基板上のノイズがマスタのFPGA自体のコンフィグレーションにも影響します．Spartan-3Aではこの点が改善され，マスタ・モード時は純粋な出力ピンとしてCCLKが構成されます．

CCLKピンは，シリーズ抵抗をピンの直近に配置するか，テブナン終端を使用するなどしてクロックの立ち上がりエッジ部にノイズが乗らないように注意する必要があります．50Ω特性インピーダンスの伝送路を想定した場合，シリーズ抵抗なら20Ω～30Ω程度が目安となります．

ユーザ・ガイド[(1)]には**図4**のようにCCLKピンにテブナン終端の例が紹介されています．CCLKがほかのFPGAやコンフィグレーションROMに分岐する場合は，スター型の配線ではなくマルチドロップにして，さらにスタブを12.5mm以内に抑える必要があります．

図4
CCLK信号の注意
シリーズ抵抗をピンの直近に配置するか，テブナン終端を使用するなどしてクロックの立ち上りエッジ部にノイズが乗らないように注意する必要がある．CCLKがほかのFPGAやコンフィグレーションROMに分岐する場合は，スター型の配線ではなくマルチドロップにして，さらにスタブを12.5mm以内に抑える必要がある．

● DONEピン

DONEピンは，コンフィグレーション中はFPGAが“L”をドライブし，完了するとハイ・インピーダンスになるオープン・ドレインのピンです．

DONEピンには330Ωのプルアップ抵抗をV_{CCAUX}電圧のレベルに付けてください．330Ωはかなり小さい値に思えるかもしれませんが，CCLKの1サイクル以内に確実にDONEピンを“L”から“H”へ遷移させる必要があることと，コンフィグレーション中の“L”ドライブ中のシンク電流値の関係で330Ωを推奨しています．

DONEピンと専用コンフィグレーションROMのCEピンを接続しておくことで，コンフィグレーション完了後DONEが“H”になるのと同時に専用コンフィグレーションROMが切り離されます．

● JTAGピン（TCK，TMS，TDI，TDO）

JTAGピンはV_{CCAUX}電源系を使用しています．このため，V_{CCAUX}が2.5Vのときは基本的にはJTAGインターフェースの電圧は2.5Vレベルで行う必要があります．V_{CCAUX}が2.5Vのときに3.3VのJTAGチェーンを組みたい場合は，TCK，TMS，TDIの各入力ピンに直列抵抗を入れて，電流制限を行う必要があります．

コンフィグレーション後のJTAGピンは，FPGA開発ツールISEのデフォルト設定では内部プルアップになります．念のため基板上でもプルアップすることをお勧めします．プルアップ抵抗値はさほど気にする必要はありませんが，一般的には4.7kΩ程度です．

● DOUTピン

図2には記載していませんが，スレーブ・モードのFPGAをカスケード接続する際に，下流のスレーブ・デバイスへコンフィグレーション・データを出力するピンです．

3．トラブル発生時のFAQ

コンフィグレーション回路は，これまで説明してきた注意点を守って設計されていれば，トラブルの原因になることはほとんどありません．しかし，コストを抑えるなどの理由で指定通りの部品を搭載できないような場合もあるでしょう．

コンフィグレーションが成功しないといったトラブルが発生したときは，まずオシロスコープを使用して以下の信号を確認します（pp.83-85のコラム「マスタ・シリアル・モードのコンフィグレーション・フロー」を参照）．

- 電源（V_{CCINT}，V_{CCAUX}，V_{CCO}）
- プログラム・ピン
- INIT_B
- DONE
- CCLK
- データ入力ピン（DIN）

特にFPGAがうんともすんともいわないような場合は，電源のランプアップが**図3**の判断基準に照らし合わせて問題ないかどうかと，プログラム・ピンが“H”に遷移しているかどうかを最初に確認します．

電源とプログラム・ピンの動作に問題がなければ何かしらコンフィグレーション回路は動いているはずです．筆者の経験では基本的にCCLKがきれいに出ているのにコンフィグレーションがうまくいかないといったトラブルのほとんどはケアレス・ミスが原因です．

● マスタ・モードFPGAのCCLKが出ない

マスタ・モードのFPGAは，電源を投入し，コンフィグレーションを行う準備が整うと，CCLKをトグルし始めます．

CCLKがトグルし始めない症状が出ている場合は，モード・ピンがマスタ・モードになっているかを確認してください．モード・ピンはINIT_Bの立ち上がりエッジで読み取られるので，INIT_Bが“L”から“H”へ遷移していることも確認します．

外部からINIT_Bを“L”に固定しているような場合や，INIT_Bのプルアップ抵抗を付け忘れている場合には，CCLKは出力されません．また，CCLKにダンピング抵抗を付けている場合は，抵抗自体が壊れていないかも念のため確認します．

● CCLKのトグルが止まらない場合

マスタのFPGAがCCLKを出力していて，コンフィグレーションROMからも何かしらのデータが出ているのに，いつまでたってもコンフィグレーションが終了しない場合があります．DONE＝“L”，INIT_B＝“H”のままとなり，あたかもコンフィグレーションをずっと行っているかの状態に見えます．

原因としては，コンフィグレーション・データの最初に入っている同期ワードをFPGAが正常に受信できていない

ことが考えられます．CCLKの立ち上がりエッジにノイズが乗って，同期ワードのビットがずれてしまうと，以降のデータをすべて読み飛ばすため，このような症状になります．

また，同期ワードの受信に成功していても，CCLKにダブル・エッジなどが発生していると，コンフィグレーション・フレームをずれた状態で認識してしまい，CRCチェック・コマンドを正常に受信できずにこの状態になる場合があります．

同期ワードの受信が成功しているかどうかを調べるには，CCLKの周波数をオシロスコープで測定するのが簡単です．同期ワードの受信に成功していれば，CCLKの周波数はユーザがISEで選択した周波数に遷移しているはずです．

同期ワードやCRCコマンドの受信に失敗する大きな原因は，上述したようにCCLKの波形が問題となる場合がほとんどです．このため，ダンピング抵抗をCCLKピンの近くに配置して波形を滑らかにしたり，FPGAと専用コンフィグレーションROMをできるだけ近くに配置するなどして対策します．CCLKはデフォルトの設定ではたかだか数MHzと低速ですが，スルー・レートは比較的高速であることと，多くのFPGAファミリにおいて，マスタ・モードのCCLKが双方向で構成されているために波形品質に注意する必要があります．

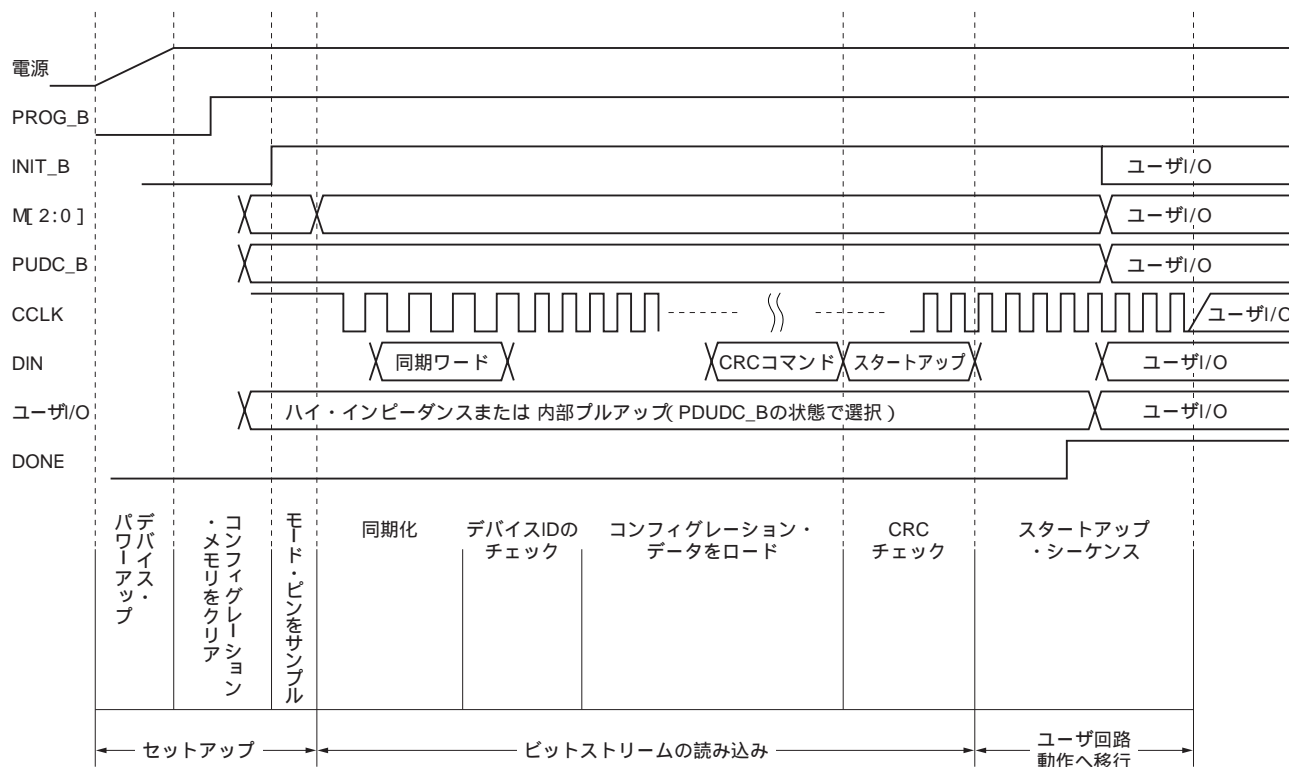
コラム　マスタ・シリアル・モードのコンフィグレーション・フロー

電源投入からユーザ回路の動作が始まるまでのフローを**図A-1**に示します．

● 電源投入からモード・ピンのサンプルまで

電源投入が完了し，プログラム・ピンが“L”の間はFPGAがコンフィグレーション・メモリの初期化ステートにとどまります．メモリ初期化中はINIT_BをFPGAが“L”にドライブします．プログラム・ピンが“H”になり初期化が完了すると，INIT_Bは外付けのプルアップにより“H”レベルに遷移し，INIT_Bの立ち上がりエッジでモード・ピンの値を読み出し，コンフィグレーション・モードが決定します．

● 同期化からCRCチェックまで

INIT_Bの立ち上りエッジでモード・ピンをサンプルした結果，マスタ・モードのいずれか（シリアル，パラレル，SPIなど）である場

図A-1　コンフィグレーション・フロー

● DONEは“L”のまま，INIT_Bも“L”になった場合

CCLKが停止し，DONEとINIT_Bが“L”の場合，CRCエラーまたはIDコード・エラーが原因でコンフィグレーションが失敗したことを表しています．

FPGA開発ツールのISEで指定したターゲット・デバイスが，実際に基板上にあるFPGAの型名と合っているかを再確認します．FPGAの型名が正しければ，CCLKの波形を確認してください．

● DONEは“H”だが何か動きがおかしい

DONEが“H”になっているので，基本的にコンフィグレーションは完了しています．

しかし，この場合でも少々注意しておきたい点があります．DONEの立ち上がり遷移は，1 CCLKサイクル以内に行われなくてはなりません．外付けプルアップ抵抗が非常に大きな値のものを使用している場合などで1サイクル以内に遷移できない場合，不安定な動作になる場合があるようです．

また，外部のプロセッサなどからスレーブ・モードでコンフィグレーションさせているような場合は，DONEが“H”になったからといって，すぐにCCLKの供給を止めてしまうとスタートアップ・シーケンスを完了できない場合

コラム　マスタ・シリアル・モードのコンフィグレーション・フロー（つづき）

合は，CCLKピンからクロックを出力し始めます．CCLKのトグル周波数は，デバイスごとにデフォルト値が決まっており，トグル開始時はこの周波数で動き出します．

クロックがトグルし始めることでPROMからデータが出力し始めます．Spartan-3 Generationファミリの場合は，ダミー・データ（FFh）がしばらく続いた後，**表A-1**に示す同期ワード（シンク・ワードともいう）を正常に受信するとコンフィグレーション・データであることが認識され，それ以降の処理を継続します．同期ワードが正常に認識されない場合は，コンフィグレーションROMから後続のデータが読み出されても，FPGAはその部分をコンフィグレーション・データとは認識せず，すべて無視します．このため，同期ワードを認識することは非常に重要なイベントです．

シリアル・モードでコンフィグレーションする場合，bitファイルのバイナリ・イメージで見るとMSBを先に読み，LSBを最後に読み込む必要があります．Xilinx社の専用コンフィグレーションROM（Platform Flash PROMなど）はLSBを最初に出力する仕様になっています．このため，**表A-2**に示すビット・スワップ（バイト・スワップと呼ぶこともある）により，MSBからLSBまでの8ビットを入れ替える必要があります．ただし，iMPACTを使用してXilinxのシリアル・コンフィグレーションROMをターゲットにbitファイルからMCSファイルを生成する場合は，ツールが自動的にビット・スワップを行ってくれます．SPIフラッシュのようなMSBを最初に出力するタイプのメモリ素子の場合は，ビット・スワップが行われないようにする必要がありますが，iMPACTでSPIフラッシュをターゲットとしてMCSを生成する場合はビット・スワップは行わないようになっています．

どちらにしても，iMPACTで正しくターゲット・デバイスを選択していれば，ビット・スワップについて何も気にする必要はありません．なお，ビット・スワップについて明示的に選択したい場合は，HEXファイルを選択することになります．

同期ワードの認識が成功した後，IDコード・チェック用のデータを受信し，実際のデバイスがターゲットとして正しいか確認されます．このため，ISEでマッピングした際のターゲット・デバイスと異なるデバイスにコンフィグレーションしようとすると，IDコード・

表A-1　同期ワードのパターン

デバイス・ファミリ	同期ワードのパターン
Spartan-3A/3AN/3A DSP	AA99
その他のVirtexアーキテクチャ・デバイス（Spartan-3/3E，Virtex-5/4 etc）	AA995566

＊ 16進数表示でビット・スワップを行っていない場合のデータ・パターンで表してある．

＊ シリアルでコンフィグレーションする場合，MSB側のビットから処理する．例えばAA99の場合はA（1→0→1→0）→A（1→0→1→0）→9（1→0→0→1）→9（1→0→0→1）のビットの順番で処理する．

表A-2　ビット・スワップの例

ビット・スワップしていないデータ	16進数表現	AA	99	55	66
	2進数表現	10101010	10011001	01010101	01100110
ビット・スワップ後のデータ	2進数表現	01010101	10011001	10101010	01100110
	16進数表現	55	99	AA	66

＊ ビット・スワップするとAAは55になる

があります．独自のコンフィグレーション・インターフェースで処理する場合は，MCSファイルやHEXファイルのデータをすべてFPGAへ送り終わった後に，DONEピンの状態を確認してOK/NGの判断をするようにしてください．データ・ファイルの最後の部分にはダミー・データが含まれており，データをすべて送り出していればスタートアップ・シーケンスを完了するのに十分な時間を取れるようになっています．

● コンフィグレーション完了直後に変なパルスが出る

以前，「コンフィグレーション後は“H”を維持するはずの信号が，DONEが上がった直後くらいに一瞬“L”が出てしまう」との問い合わせを受けたことがありました．

問題となった出力信号は，リセット系の信号を作っているI/Oでした．このため，パルスを出力すると外部のリセット信号をアサートしてしまい，単純な回路構成ながらかなり問題視された記憶があります．

コンフィグレーションの最後の工程であるスタートアップ・シーケンスにおいて，FPGA設計ツールISEのデフォルトの設定では，DONE→GTS→GWEの順番でイベントが発生します．DONEを開放して外部にコンフィグレーションの完了を通知した後に，出力ピンがドライブを開始

チェックの段階で中断するので，デバイスが破損するようなことはありません．

IDチェックも成功すると，ISEのGenerate Programming Fileのプロパティで設定したCCLKの速度へ遷移します．デバッグ時にはCCLKの周波数がデフォルト周波数から変化したかどうかを確認すれば，この段階まで処理が進んだかどうかチェックできます．

その後はコンフィグレーション・データを読み込み続け，CRCチェックのコマンドとCRCの期待値をFPGAに送り込み，FPGA内部でCRCの計算を行い，期待値と一致していれば，最終段階のスタートアップ・シーケンスへと進みます．

● スタートアップ・シーケンス

スタートアップ・シーケンスは，コンフィグレーション動作からユーザ回路動作へ移行する8フェーズのシーケンスです．このシーケンスの中で行われる主な処理は，DONEピンの開放（“L”のドライブをやめる），ユーザ回路の出力ピンのドライブ開始，フリップフロップやRAMといった同期エレメントへの書き込みの開始などです．デフォルトの設定では**図A-2**のタイミングで各イベントが発生します．また，同期エレメントの初期化に使用されるGSR（Global Set Reset）のアサートやDCMのロックの開始は，スタートアップ・シーケンスの最初のフェーズで行われます．

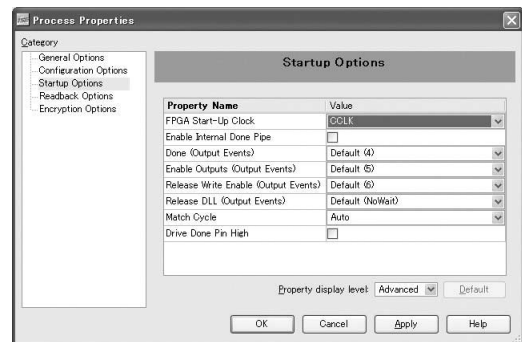
スタートアップ・シーケンスで使用するクロックはISEの中で**図A-3**のプロパティ設定ウィンドウで，CCLK，JTAGクロック，ユーザ・クロックの三つの中から選択します．基本的にJTAGからコンフィグレーションするときはJTAGクロックを選択し，それ以外のときはCCLKを選択します．

スタートアップ・クロックの選択を間違えると，スタートアップ・シーケンスが実行できないため，コンフィグレーションも完了できません．しかし，iMPACTでコンフィグレーションROMやFPGAへ書き込む際に，一般的に使用されるクロック系と異なるものを選択している場合は，Warningを表示してクロック系の変更をしてくれます．

スタートアップ・シーケンスが最後まで行われることで，ユーザ回路動作への移行が完了となります．

● ユーザ回路の動作開始

図A-1に示す信号の中で，ユーザI/Oとして使用可能なものがいくつかあります．それらをユーザI/Oとして使用していない場合は，通常の未使用I/Oピンの処理が行われ，デフォルトでは内部プルダウンが働きます．このためINIT_Bピンのような基板上にプルアップが付いているもので，コンフィグレーション後にユーザI/Oとしては使用しない場合は，個別にピンをハイ・インピーダンス状態にするなどの処理をユーザ回路の一部で行うと，リーク電流を抑えられます．

図A-2　Spartan-3 Generationのデフォルト・スタートアップ・シーケンス

図A-3　スタートアップ・シーケンスのオプション設定

し，さらにその後に内部のフリップフロップへの書き込みが行えるようになります．フリップフロップの初期化はスタートアップ・シーケンスの先頭ですでに行われており，このとき問題となったフリップフロップは‘ 0 ’に初期化されていました．フリップフロップへの入力データは1で，クロックも来ているにもかかわらずGWEのアサートがGTSより後になっているため，一瞬だけ初期値の‘ 0 ’が外部に出力されることが原因でした．

GWEがGTSより前に発生するようにスタートアップ・シーケンスを変更することで，コンフィグレーション直後に“ L ”パルスが出ることはなくなりました．念のため記しておきますが，これはツールや基板の不具合ではなく，すべて既知の動作でした．

● **コンフィグレーション後にINIT_Bが“L”になる**

DONEが“ H ”になった後，INIT_Bが“ L ”になってしまう場合があります．

コンフィグレーションが完了（DONEが“ H ”）した後，Spartan-3 GenerationのFPGAでは，INIT_Bピンはユーザ I/Oとして使用できるようになります．このときユーザ回路でINIT_Bピンを使用していない場合は，未使用ピンの扱いとなります．未使用ピンは，FPGA設計ツールISEのデフォルト設定では内部プルダウンです．このため，コンフィグレーション完了後にINIT_Bが“ L ”に遷移するのは，多くのケースでまったく問題ない動作です．

もちろん，未使用ピンの処理を内部プルアップに設定すれば，コンフィグレーション後にINIT_Bをユーザ回路で使用していない場合に“ H ”が現れるようになります．

補足すると，INIT_Bには基板上に4.7kΩのプルアップ抵抗が付いているので，内部プルダウン抵抗と基板上のプルアップ抵抗がつながることになり，結果としてINIT_Bは完全にはGNDには落ちません．これでFPGAが破壊に至ることはないのですが，気になるようでしたらユーザ回路でINIT_Bピンが“ H ”またはハイ・インピーダンスになるように設計しておくとよいでしょう．

参考・引用*文献

(1) Spartan-3 Generation Configuration User Guide，Xilinx，May 23, 2007.
(2) Virtex-5 FPGA Configuration User Guide，Xilinx，July 31, 2007.
(3) Virtex-4 Configration Guide，Xilinx，Aug. 8，2007.

やすい・けん
アヴネット ジャパン（株）

<筆者プロフィール>
安井　健． 2001年9月，アヴネットジャパン（当時メメックジャパン）入社以来，FAEとして活動．それ以前は通信系の電気メーカで電気系CADツールの社内サポートなどを行っていた．